

ROVANIEMI UNIVERSITY OF APPLIED SCIENCES

SCHOOL OF TECHNOLOGY

Degree Programme in Information Technology

Thesis

ARDUINO SHIELD DESIGNS FOR ELECTRONICS LABORATORY STUDIES

Jarno Ruokamo

2012

Supervisor: Tauno Tepsa

Approved _____2012_____

The thesis can be borrowed.

Author	Jarno Ruokamo	Year	2012
Subject of thesis	Arduino Shield Designs for Electronics Laboratory Studies		
Number of pages	28		

The aim of the thesis was to create a collection of shields to be used with the Arduino microcontroller platform that replaces or supplements the features of the existing platform used by the electronics laboratory.

The thesis covers the design and construction processes and explains the general properties of the shields. The work started with the designing of the shields using an open source software suite KiCad which contains all the tools required to design PCB schematics and board layouts.

In total, five shields were designed. Four of them were made, three with the PCB router in the electronics laboratory and the remaining one was ordered from Lahden Autec due to the complexity of the layout.

Key words kicad, pcb, router, arduino
Special remarks This work contains schematics of the shields.

Tekijä	Jarno Ruokamo	Vuosi	2012
Työn nimi	Arduino Shield Designs for Electronics Laboratory Studies		
Sivu- ja liitemäärä	28		

Opinnäytteen tavoite oli luoda kokoelma Arduinoon liitettäviä lisäosia, jotka korvaavat tai parantavat nykyisin elektroniikan laboratoriossa käytössä olevan alustan ominaisuuksia.

Tämä opinnäytetyö sisältää lisäosien suunnittelun, rakennuksen ja niiden perusominaisuuksiin tutustumisen. Opinnäytetyö alkoi lisäosien suunnittelulla vapaan lähdekoodin ohjelmistolla KiCad. Se sisältää kaikki piirilevyjen suunnitteluun vaaditut työkalut.

Työn tuloksena suunniteltiin viisi piirilevyä, joista neljä rakennettiin. Kolme käyttäen elektroniikan laboratoriossa olevaa jysintä ja yksi käyttäen Lahden Autecin apua piirilevyn monimutkaisuuden takia.

Avainsanat
Muita tietoja

kicad, pcb, jysin, arduino
Työ sisältää piirilevykaavat.

CONTENTS

1 INTRODUCTION.....	2
1.1 AVRKIT.....	2
1.2 GOAL OF THE THESIS	4
1.3 HISTORY OF ARDUINO	5
2 COMPETITORS OF ARDUINO	7
2.1 TEXAS INSTRUMENTS LAUNCHPAD MSP430	7
2.2 TEXAS INSTRUMENTS LAUNCHPAD STELLARIS	7
2.3 STM32 DISCOVERY	7
2.4 FREESCALE FREEDOM.....	7
3 DESIGN PROCESS.....	8
3.1 KICAD	8
3.2 EESHEMA	8
3.3 CVPCB.....	10
3.4 PCBNEW	10
3.5 GERBVIEW	11
4 CONSTRUCTION PROCESS	13
4.1 CIRCUITCAM.....	13
4.2 BOARDMASTER.....	13
5 SHIELDS.....	16
5.1 OVERVIEW	16
5.2 SHIELD FOR LCD SCREENS	17
5.3 SHIELD FOR CHARLIEPLEXED 6X5 LED MATRIX.....	18
5.4 SHIELD FOR 6 MULTIPLEXED 7-SEGMENT DISPLAYS	19
6 LIBRARIES.....	20
6.1 LCD	20
6.2 MAXIM	21
7 CONCLUSION	23

LIST OF FIGURES

FIGURE 1. AVRKit	2
FIGURE 2. USB PROGRAMMER MODULE	2
FIGURE 3. STRIP OF EIGHT RED LIGHT EMITTING DIODES.....	3
FIGURE 4. FOUR PUSH BUTTONS AND FOUR RED LIGHT EMITTING DIODES	3
FIGURE 5. SERIAL CONNECTION BY AN FTDI CONVERTER.....	3
FIGURE 6. FOUR POTENTIOMETERS FOR VARIABLE INPUT	3
FIGURE 7. SCREW TERMINALS FOR ANALOG INPUT SIGNALS	4
FIGURE 8. SIMPLE BREADBOARD WITH A DS18B20 DIGITAL 1-WIRE THERMOMETER.....	4
FIGURE 9. WINTeK 27x4 LCD SCREEN WITH A RIBBON CABLE CONNECTOR	4
FIGURE 10. KiCAD PROJECT MANAGER.....	8
FIGURE 11. EXAMPLE EESchema VIEW.....	9
FIGURE 12. EXAMPLE CVPcb VIEW	10
FIGURE 13. EXAMPLE PcbNEW VIEW	11
FIGURE 14. EXAMPLE GERbVIEW VIEW	12
FIGURE 15. EXAMPLE CIRCUITCAM VIEW	13
FIGURE 16. EXAMPLE BOARDMASTER VIEW	15
FIGURE 17. LCD SHIELD 1: FOR 27x4 LCD WITH A FLAT FLEX CABLE CONNECTOR.....	16
FIGURE 18. LCD SHIELD 2: FOR AN 16x2 LCD	16
FIGURE 19. CHARLIEPLEXED 6x5 LED MATRIX.....	17
FIGURE 20. SIX MULTIPLEXED 7-SEGMENT DISPLAYS.....	17
FIGURE 21. CHARLIEPLEX SETUP.....	19

1 INTRODUCTION

1.1 AVRKit

The current microcontroller platform in use at the electronics laboratory is a kit made by the Seinäjoki University of Applied Sciences (SUAS). The kit consists of a motherboard containing a megaAVR microcontroller, either an ATmega16 or an ATmega32 and provides six connectors for programming and connecting the daughter boards.

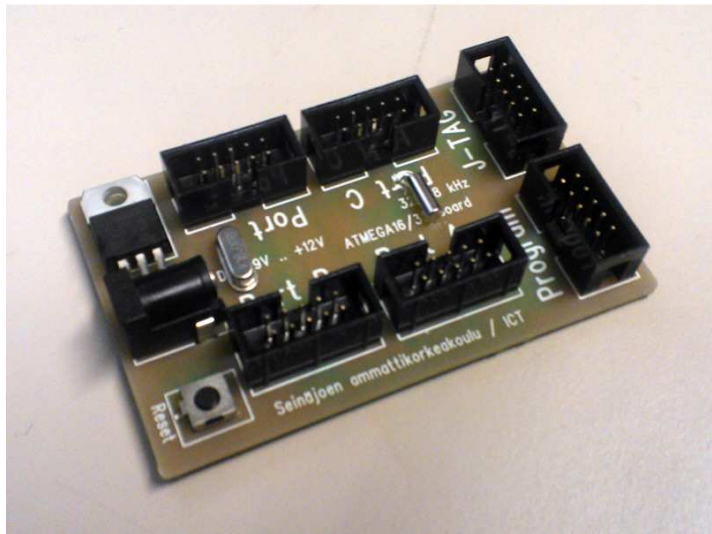


Figure 1. AVRKit

Two of the ports are special, one for JTAG debugging of the running code in the microcontroller and a second one for connecting the USB programmer. The programmer is a board with an ATmega8 with a connector to the motherboard and an USB-B connector. It is programmed with the *usbasp* firmware, which acts as an interface for the programming utilities on the computer and the target microcontroller on the motherboard.



Figure 2. USB Programmer Module

The remaining four ports are for the digital input/output ports available in the microcontroller. They can use the daughter boards which were provided by SUAS with the AVRKit as a package. Figures 3, 4, 5, 6 and 7 show modules made by SUAS. Modules shown in figures 8 and 9 were made by the electronics laboratory.

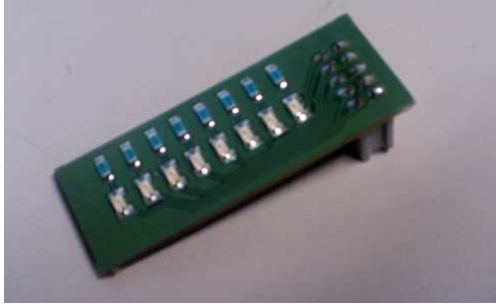


Figure 3. Strip of Eight Red Light Emitting Diodes

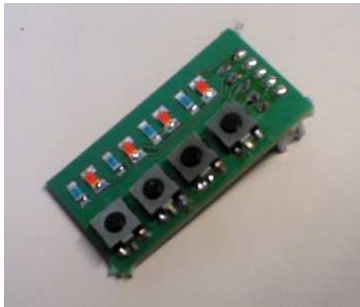


Figure 4. Four Push Buttons and Four Red Light Emitting Diodes



Figure 5. Serial Connection by an FTDI Converter

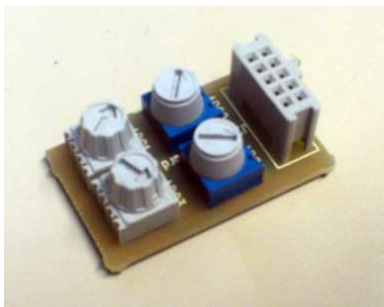


Figure 6. Four Potentiometers for Variable Input

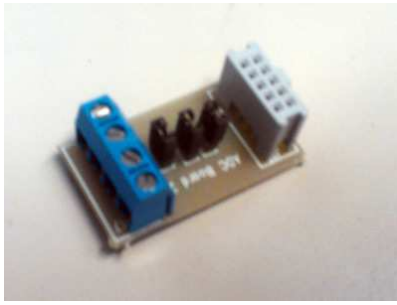


Figure 7. Screw Terminals for Analog Input Signals

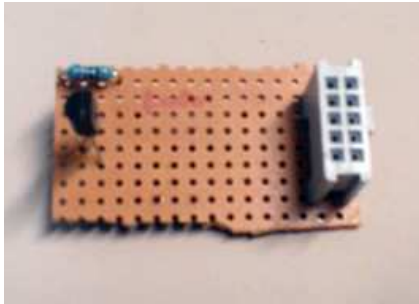


Figure 8. Simple Breadboard With a DS18B20 Digital 1-Wire Thermometer



Figure 9. Wintek 27x4 LCD Screen With a Ribbon Cable Connector

1.2 Goal of the Thesis

The aim is to create a collection of shields which match or supplement the features of the existing microcontroller board. By using the widely available Arduino board, any student can buy their own Arduino and loan the shields from the Rovaniemi University of Applied Sciences (RAMK) for home use. In the electronics laboratory the students use boards supplied by RAMK. Additionally, if ever a project may require more than the 20 input/output pins that are provided by the standard Arduino board, the Mega version can be used instead providing a total of 70 input/output pins.

1.3 History of Arduino

Arduino begun in 2005 when professor Massimo Banzi at the Interaction Design Institute in the city of Ivrea, Italy, wanted to make it easier for students to learn about electronics. Give students the opportunity to discover things themselves, rather than listening to nothing but theory. For this he found help from David Cuartielles, a visiting engineer from Malmö University and together they decided to make a microcontroller that would be more accessible to use. Two students were chosen to write the software for the device. Electrical engineer Gianluca Martino was asked to make an initial batch of 200 microcontrollers. The name Arduino was taken from a historical character, Arduin of Ivrea. (Lajart 2009.)

The first Arduino made was called “Arduino Serial” and included an ATmega8 with a direct RS-232 connection to the microcontroller and all through-hole components. Afterwards a 2.0 version and a single sided version for etching for hobbyists was designed. Versions after these were all USB with an FTDI converter. After USB v1.0 and v2.0, Arduino Extreme was released which increased the amount surface mount components. Arduino Nuova Generazione switched to a simpler USB converter and moved from ATmega8 to ATmega168. Improvements continued with Diecimila, Duemilanove and Uno by replacing the FTDI converter with an ATmega8U2 which contains an USB controller and upgrading the microcontroller to ATmega328. The latest version of the basic Arduino is Leonardo, which moves from the easily interchangeable socketed microcontroller to a surface mounted one that requires soldering to change and eliminates the converter completely by upgrading to an ATmega32U4 which contains an USB controller thus simplifying the design greatly. (Massimo 2008.)

Speciality versions have also been made. Smaller versions such as Mini and Nano allow smaller installations. Larger versions with ATmega1260 and ATmega2560 for greater expandability and more pins. Remote versions are also available in the form of Arduino BT for Bluetooth connectivity and Arduino Ethernet for LAN connection which also has support for Power over Ethernet if an add-on module is installed on the board. Arduino Lilypad is wash proof and has been designed for flexibility and sewing into clothing.

Shields can be plugged to the pins available on the Arduino and official shields include Wi-Fi, Ethernet, Wireless, Motor and Proto for prototyping. Wireless shield comes with either SD slot or prototyping space and has headers for connecting modules which use the XBee footprint. Motor allows control of 2 DC motors. Anyone anywhere can make shields themselves if they can produce printed circuit boards. (Massimo 2008.)

2 COMPETITORS OF ARDUINO

2.1 Texas Instruments Launchpad MSP430

Texas Instruments in 2010 released a low cost MSP430 based microcontroller platform. The Launchpad development tool is an embedded USB-powered emulation tool. It supports all 14 and 20 pin DIP socket MSP430 based Value Line and Spy Bi-Wire devices. It also supports modules called BoosterPacks which connect to the pins on the Launchpad and provide features such as capacitive touch, RGB led, wireless communication and many more. (Texas Instruments 2010.)

2.2 Texas Instruments Launchpad Stellaris

Texas Instruments in September 2012 announced a low cost ARM© Cortex™-M4 based microcontroller platform. It expands the previously released MSP430 Launchpad platform by doubling the number of pins and increasing the performance by moving to an ARM based platform. It supports an expanded BoosterPack XL modules for even greater expandability. (Texas Instruments 2012.)

2.3 STM32 Discovery

STMicroelectronics in September 2010 released a low cost ARM© Cortex™-M3 based microcontroller platform. A 24MHz microcontroller with 48 input/output lines and large number of features with a free development software. (STMicroelectronics 2012.)

2.4 Freescale Freedom

Freescale in September 2012 released a low cost ARM© Cortex™-M0 microcontroller platform. A 48MHz microcontroller with embedded peripherals on the board, capacitive touch slider, RGB led and an accelerometer. It also has pin compatibility with Arduino R3 signals. (Freescale Semiconductor 2012.)

3 DESIGN PROCESS

3.1 KiCad

Design of the shields is performed with the KiCad EDA Software Suite. EDA standing for Electronic Design Automation. KiCad consists of five major components which are the following.

- kicad - Manages overall project of a board design.
- eeschema - Schematics building and editing tool
- cvpcb - Tool for selecting a footprint for the components on the board
- pcbnew - PCB Layout building and editing tool
- gerbview - Gerber file viewer

KiCad is the main project manager application. It can archive projects and open archived projects, change the language of the applications and used for starting all the other applications.

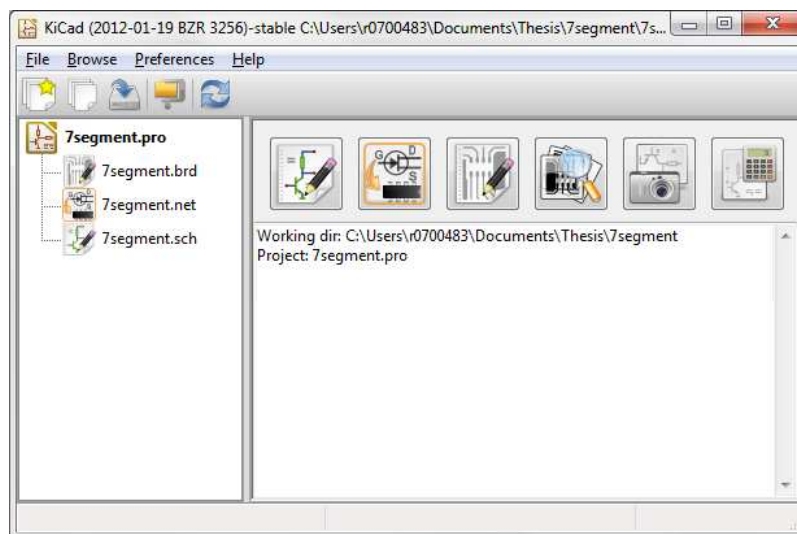


Figure 10. KiCad Project Manager

3.2 Eeschema

Eeschema is the main schematics editing program. When started for the first time, it creates a default A4 size schematics sheet. To start making a schematic, components must be added first. By clicking Place a component - button, a dialog is opened where the user can either directly input the name of a component if it is known or can search all the available libraries for the component. If a component does not exist, it is possible to create new

components using the Library editor. An alternative is to convert components from other EDA software. Existing components can be found online and are often a better choice because they more likely to have been tested by other people. With the components placed on the sheet, connections are made by using wires or busses. Labels can be used to define pins which all connect to each other and “No connect flags” can be used to define a pin which must not be connected to anything at all. Finally power ports are used to connect input voltages, output voltages and ground.

When the design has been finished by adding components and connecting them fully, the components on the design must be annotated. When placing components such as resistors, each component is placed with an initial name of R?. When the annotation is performed, each component gets a unique identifier replacing the ‘?’ and becoming such R1, R2 and increasing for each component. After this, an ERC, electric rules check, is good to perform to ensure all pins are either connected correctly or defined as no connect. When ERC passes successfully, it is time to generate a net list. The net list defines a list of nets and the pins which connect to these nets. If a net has been named it will use that name, otherwise it will be designated a random name.

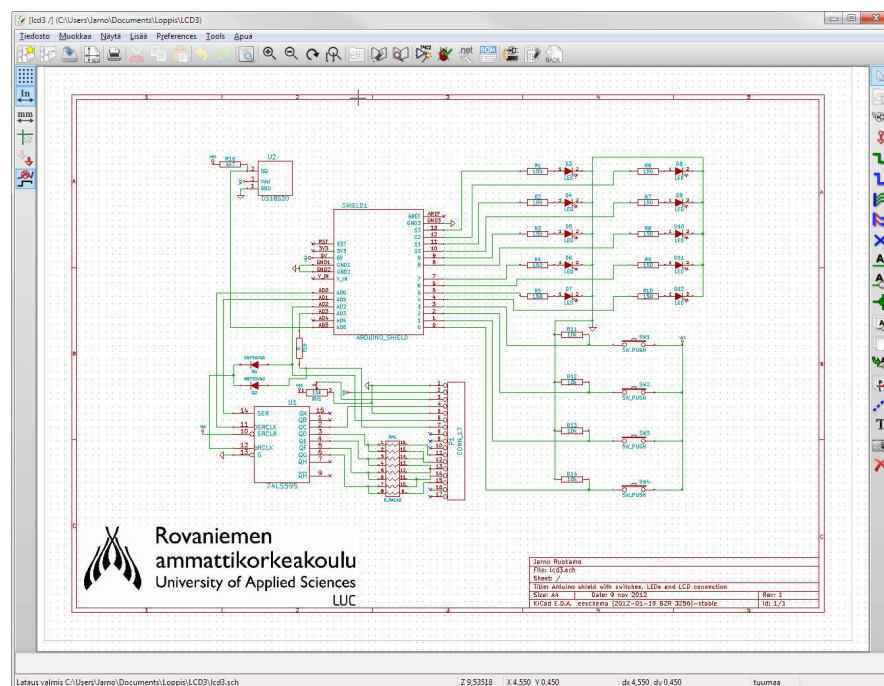


Figure 11. Example Eeschema View

Spectra Session file which pcbnew can import and use to populate connections made by FreeRouter. After the connections are completed, a DRC, design rules check, is made. This ensures the connections are spaced correctly and far enough from pads and displays any unconnected pins. Lastly the board edge is drawn on the edge layer and the layout is final.

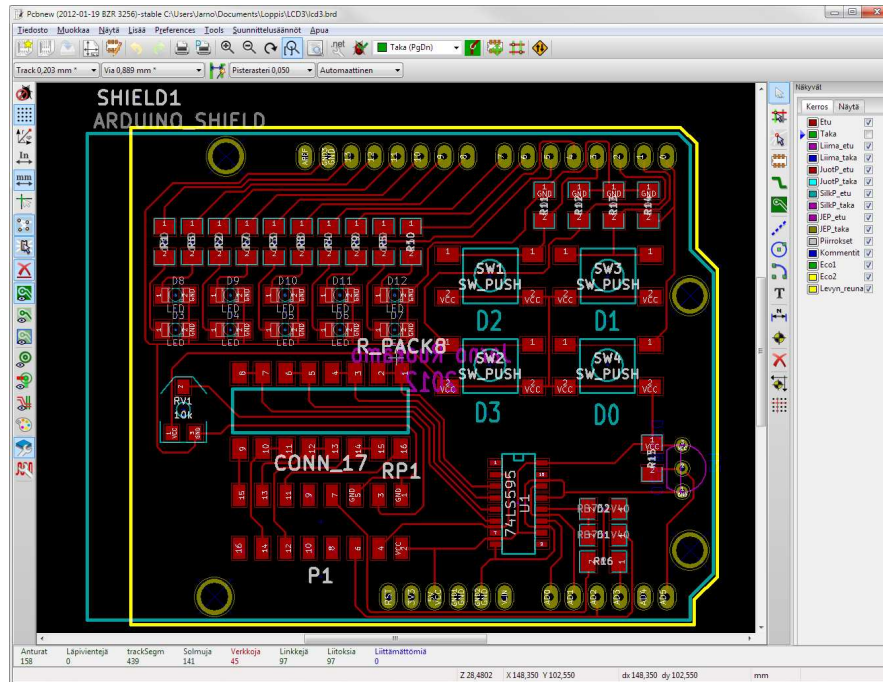


Figure 13. Example Pcbnew View

3.5 Gerbview

GerbView is used to view generated Gerber files to ensure that all components, pads, tracks and vias are placed correctly for the actual production of the board.

4 CONSTRUCTION PROCESS

4.1 CircuitCAM

With the design and layout complete and Gerber files exported, it is time to start producing the board. For this phase, I used a guide made by a student in 2009 for the LPKF Protomat S62.

After starting CircuitCAM, first job is to import the Gerber files. CircuitCAM then asks which layer the files belong to, bottom and top copper layers, board outline and drilling holes. Next job is to create the board cutting area. Board outline is created during the layout and then imported from the Gerber files and CircuitCAM generates a CuttingOutside layer from it with the Contour Routing tool. After this, select the copper layers and click “Insulate All Layers”. This generates the paths for the Universal Cutter bit to travel to create all the tracks and separate the pads. Last is to export the board for BoardMaster.

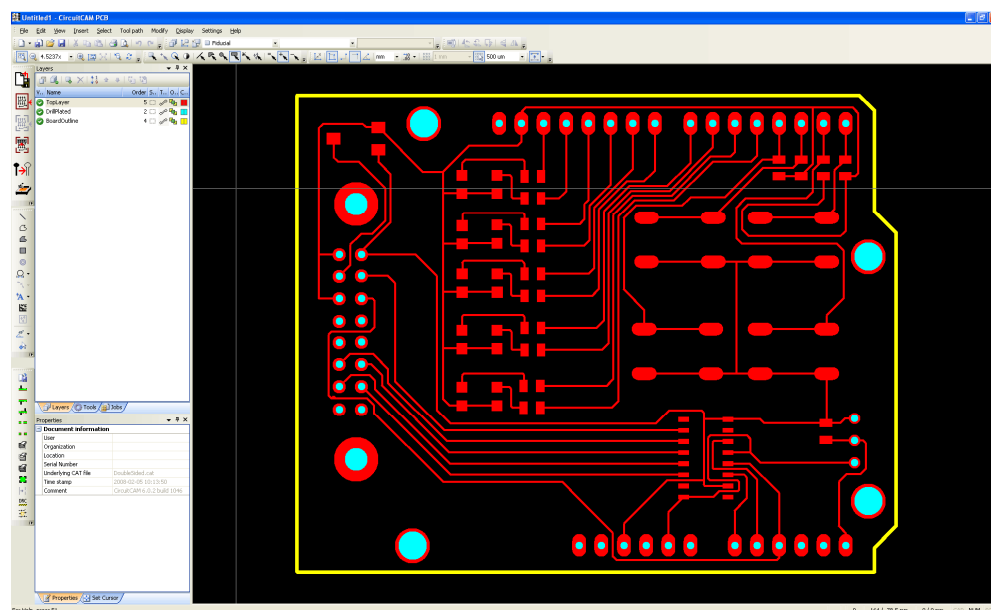


Figure 15. Example CircuitCAM View

4.2 BoardMaster

You begin by turning on the router because BoardMaster requires a connection with the router while starting up. The default template is first copied to your project folder and opened. Next the router is parked, bringing the pad out front allowing access to it and then the blank board is put on it. Before the board is started, fiducial holes are created. These are 4 1,5 mm

drill holes made in very specific locations. This way the board can be flipped, moved or removed for inspection and the router can with the camera look for these points and precisely reposition entire board. With the fiducials created, your own job can be imported. When the job is imported, it should be placed along the edge of the board to allow maximum board reusage.

Next is important to check that you have all the tools needed for the job. By opening the toolbox, it will show all the current tools in the router and also which tools are needed to complete the board. If a tool that is needed for the job is not in the toolbox, the router is parked to get access to the tools and then unneeded tools can be removed and required put in their places.

BoardMaster needs to be told which tools are in which positions, if tools are incorrectly placed they will break and possibly do damage when used. Once all the tools are correctly placed, the actual milling can take place.

1. MarkingDrills: A Universal Cutter is used to make small marking holes.
2. DrillingPlated: Drill holes for the through-hole components and vias are made.
3. MillingBottom: The tracks and pads are routed using a Universal Cutter for the bottom layer, this layer is for two sided boards and single layer boards with only through-hole components.
4. ReadTop: If the board is a two-sided board, at this point it is flipped and this phase finds the fiducial holes and adjusts the remaining phases to line up properly.
5. MillingTop: The tracks and pads are routed using a Universal Cutter for the top layer, if the board is single layer and only uses surface mounted components, this phase is the only milling phase and MillingBottom is not performed.
6. CuttingOutside: The finished board is cut from the rest of the board by using a large contour router.

With all the phases complete, the router can be parked and the finished board taken. All that remains is to clean the finished board, check for any short circuits and possibly treat it with a protective chemical. During the

milling phases, before starting the phase, BoardMaster asks for tool adjustment. This is a crucial step to take since having the correct depth means that the router can correctly adjust the depth of the Universal Cutter since its cutting width varies from 0,2 mm to 0,5 mm depending on how deep the bit is sunk. The adjustment is started by putting the router head to a point outside the borders of the projects placed on the board. With the motor running, the bit is lowered on the surface of the board. By moving the head while the spindle is on, it carves a groove on the board and after this the router is parked which moves the board forward for inspection. Using the microscope, the groove can be inspected with the scale included in the eyepiece, to check if the cut is exactly 0,2 mm. If it is not exactly that, the tool guard shroud can be lowered or raised to change the width.

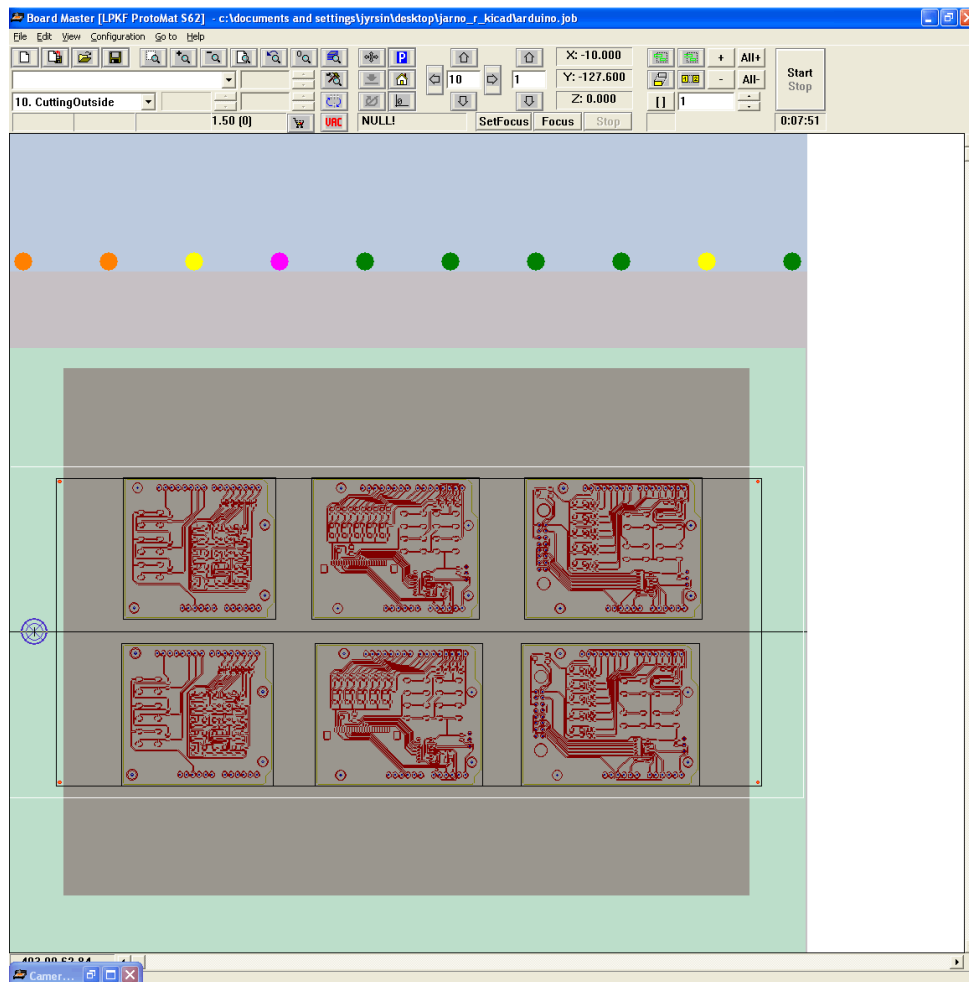


Figure 16. Example BoardMaster View

5 SHIELDS

5.1 Overview

Initially in the project three kinds of shields with two versions of one of them was selected. At the end of the project a third version of the one was designed for a total of 5 shields. Figures 17, 18, 19 and 20 show finished boards that were made during the thesis.

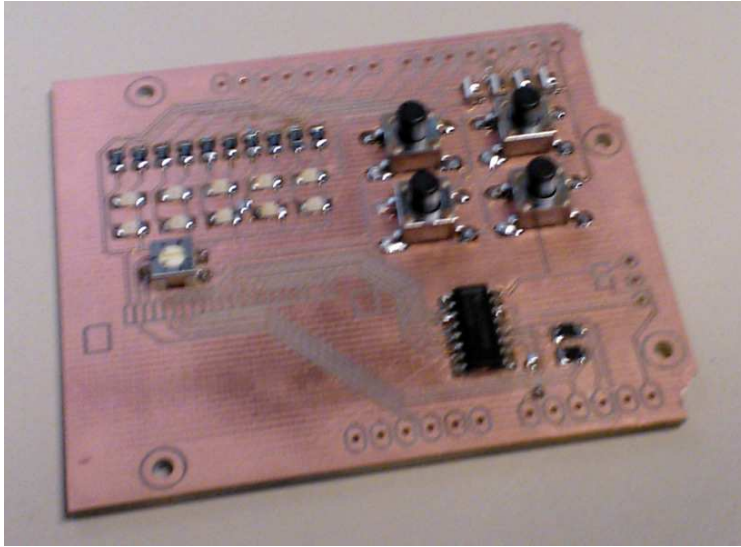


Figure 17. LCD Shield 1: For 27x4 LCD With a Flat Flex Cable Connector

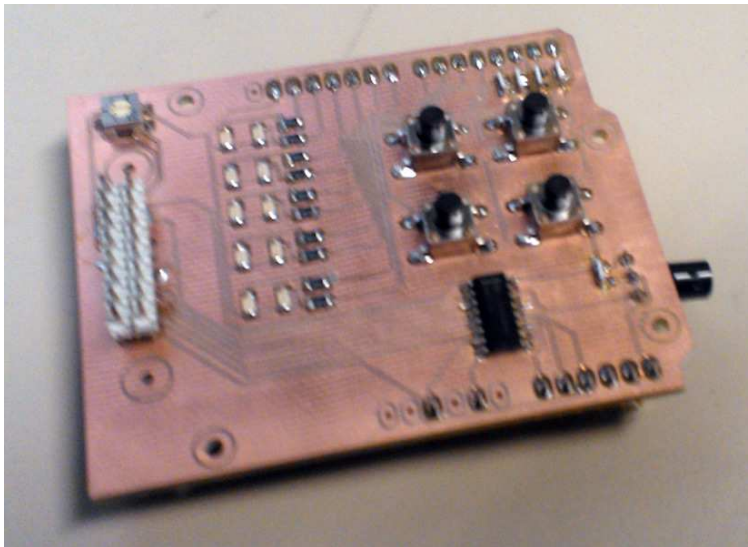


Figure 18. LCD Shield 2: For an 16x2 LCD

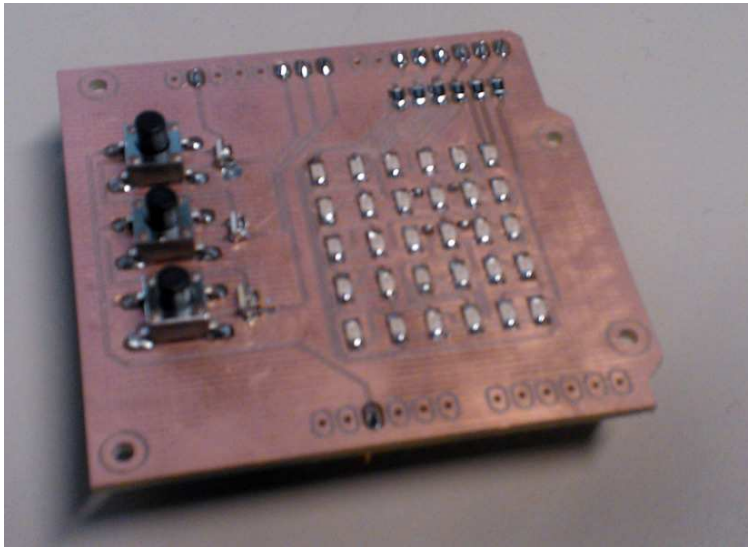


Figure 19. Charlieplexed 6x5 LED Matrix

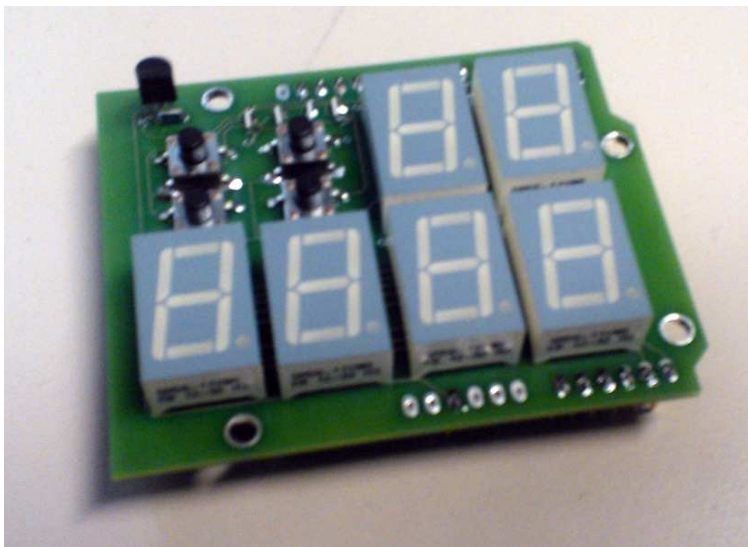


Figure 20. Six Multiplexed 7-segment Displays

In addition the four constructed, a third version of the LCD Shield was designed but not constructed because I came up with the design very late in the process. The third version is a hybrid of the first two versions and can accept both the 27x4 and 16x2 LCDs provided that the flat flex cable has in the large screen has been replaced with an IDC ribbon cable. This is due to DIP switch which changes the arrangement of the data bus.

5.2 Shield for LCD Screens

The LCD shield was designed in multiple ways to try different designs and ultimately choosing one as the shield mainly used for most of the electronics laboratory studies. The LCD shields share large amounts of design with only

different orientations and some small changes. All three have 10 LEDs, 4 push buttons and a digital thermometer. In all the designs, the push buttons use pins 0-3, LEDs use pins 4-13 and the thermometer uses pin A5. Both LCDs require a minimum of 6 pins and with only 5 pins left for use the shields use a shift register to increase the amount of pins.

Both of the LCD screen use a 4 bit parallel data bus, but the location of the pins differ. Because the 27x4 contains two controllers, it has two chip select pins and this moves the locations of the data pins up by one. This is the reason for two different boards because the data bus is in different locations and having one or two chip select lines. The 27x4 LCD screen has the data bus on pins 12-15 and the 16x2 has it on pins 11-14.

After I had designed and produced the original two versions, I came up with a third alternative which allows to use of both boards with a single shield by using a series of switches to shift the data bus up by one. Both of the chip selects are connected, and thus used as needed. The 4 data lines are connected to a DIP switch, the data lines from the shift register as follows: D4 to switches 1 and 2, D5 to switches 3 and 4, D6 to switches 5 and 6 and D7 to switches 7 and 8. Switch 1 connects to pin 11, switches 2 and 3 connect to pin 12, switches 4 and 5 connect to pin 13, switches 6 and 7 connect to pin 14 and switch 8 connects to pin 15. With this configuration, it is possible to set the data bus on pins 11-14 by connecting the uneven switches, and set it on pins 12-15 by connecting the even switches.

5.3 Shield for Charlieplexed 6x5 LED Matrix

The Charlie shield is a matrix of LEDs. A common way to control a matrix is with multiplexing, this design uses a pin for each row and column of the matrix, with the driving voltage coming from the rows and an alternating column drains the voltage. Charlieplexing is a technique invented by Charlie Allen and utilises tri-state pins used in current microcontrollers and the nature of semiconductors, current flows in one direction but is blocked in the other direction. Tri-state pins are pins which can work in three different states, voltage source, voltage drain and high-impedance. The amount of LEDs possible with this technique is dependent on the number of pins, with n pins, the element amount is $n(n-1)$.

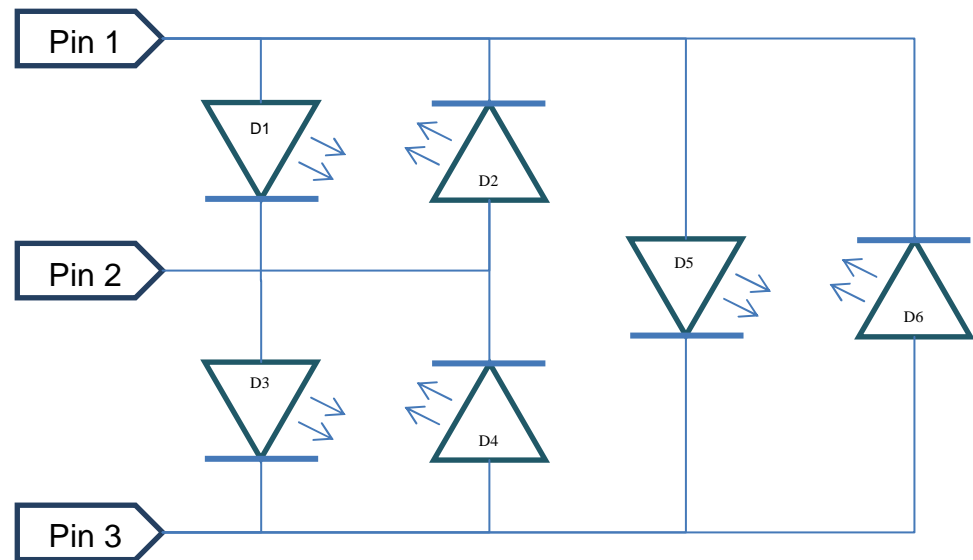


Figure 21. Charlieplex Setup

In this setup as an example, is a 6 LED charlieplex utilising 3 pins. By using tri-state pins, it is possible to illuminate for example D6, turning Pin 3 to a voltage source, Pin 2 as high-impedance and Pin 1 as voltage drain. Current travels from Pin 3 to Pin 1 and lights D6. Current cannot travel through D4 and D2 because the voltage drops is not sufficient to break through both LEDs. Pin 2 in high-impedance is effectively disconnected from the entire circuit and does not affect the circuit. (Maxim Integrated 2003)

5.4 Shield for 6 Multiplexed 7-segment Displays

This shield provides 6 multiplexed 7-segment displays, 4 push buttons and a digital thermometer. The 7-segment displays are connected so that pins A0 to A5 are connected to a transistor which is also connected to the Arduinos +5V pin and the common anode of each 7-segment display. Pins 0-7 are connected to cathodes of the displays. Each of those pins connects to one specific segment of all the displays and is used to drain the voltage provided by the transistor. The transistors are alternated so that one display is on at the time. Pin 9 is connected to the digital thermometer and pins 10-13 are connected to the push buttons.

6 LIBRARIES

6.1 LCD

In addition to the standard Arduino libraries, the LCD display shields require the use of *New LiquidCrystal* library made by Francisco Malpartida which expands the existing LCD library available in Arduino by allowing the use of a shift register or an I2C expander which help reduce the number of pins required to drive an LCD display. The implementations in both the FFC and DIL use a setup called *three wire latch shift register*. The difference is that the FFC has two enable lines for the upper and lower portions of the display while the DIL has only one. The usage is shown below. (Malpartida 2012.)

```
#include <LCD.h>
#include <LiquidCrystal_SR.h>

LiquidCrystal_SR iLCD(Data_Pin,Clock_Pin,Enable_Pin);

void setup()
{
    iLCD.begin(LCD_Width,LCD_Height);
}
```

The two statements at the beginning include the library into the project so that it can be used. The next line creates an instance of the `LiquidCrystal_SR` class through which the LCD is controlled. And the line inside `setup` defines the size of the LCD display so text can be formatted correctly. For the DIL setup it would be 16 wide and 2 high. The bigger FFC display requires some changes to the code. Instead of one enable line, the larger display requires two enable lines for two upper lines and two lower lines of text. The changes are shown below.


```

LiquidCrystal_SR i1LCD(Data_Pin,Clock_Pin,E1_Pin);
LiquidCrystal_SR i2LCD(Data_Pin,Clock_Pin,E2_Pin);

void setup()
{
  i1LCD.begin(LCD_Width,LCD_Height);
  i2LCD.begin(LCD_Width,LCD_Height);
}

```

To control the upper and lower lines, two instances of the class are required because the display has two enable lines. To control the upper lines class i1LCD is used and for the lower lines class i2LCD is used. The width for the display in question is 27 and height is 4 so for each of the instance should have a height of 2.

The class provides simple commands for writing data to the LCD display.

- `iLCD.home();` // Sets the location of the cursor to the upper left corner.
- `iLCD.setCursor(x,y);` // Set the location of the cursor to the xth row and yth line.
- `iLCD.print("Hello");` // Outputs the text sent as argument at the current cursor position

6.2 Maxim

For the Maxim Integrated DS18B20+ Programmable Resolution 1-wire Digital Thermometer, two additional libraries are needed. One Wire library for low level access to the device, and Dallas Temperature Control library by Miles Burton for easier access to the temperature data. The usage is shown below.

```
#include <OneWire.h>
#include <DallasTemperature.h>

OneWire oneWire(One_Wire_Pin);
DallasTemperature sensor(&oneWire);
float result;

void setup()
{
    sensor.begin();
}

void loop()
{
    sensor.requestTemperatures();
    result = sensor.getTempCByIndex(0);
}
```

The two statements at the beginning include the library into the project so that it can be used. The next line creates an instance of `OneWire` which is then referenced to an instance of `DallasTemperature` through which the temperature is obtained. The result variable is used to store the temperature received from the thermometer. Line `sensor.begin()` initializes the thermometer and sets the precision of the temperature at a default value. Line `sensor.requestTemperatures()` sends a command to all devices on the bus for temperature data. Finally the last line saves the temperature of the first device on the bus to the variable `result`. (Burton 2012.)

7 CONCLUSION

The thesis idea was first thought when the supervisor of this thesis was thinking of alternatives for the current kit. One of them was Launchpad simply because it was less than 4€ for each board. But he decided that the small number of pins made it unsuitable. The next choice was Arduino, for its large hobby community and good support.

Arduino uses shields for expandability and they could be used to replicate the current kit as well as possible. The LCD shields were an obvious choice, since the LED strip and the push buttons were the most used modules. The LED matrix was something very interesting to implement and the 7-segments was just extra.

The first task was to design all the boards and to find the shield footprint. After that the designs were simple and easy to implement, but the layout was most tedious. The easiest boards that can be made on the RAMK router are single-sided boards. Thus extra time was used to design all the boards single-sided.

Finally all boards except the 7-segment board were done single-sided. Thus the double-sided 7-segment board was produced by Lahden Autec. With the boards finished and the parts having arrived, soldering all the components was started. During the soldering, it was discovered that the 0805 sized components chosen for the initial designs were too small for easy handling. The supervisor recommended redesigning all the boards using the 1206 sized components.

The final boards use the 1206 sized components. During the project, the schematics took two weeks but optimizing the layout took a month. Getting the tracks of the layout to occupy only one side of a printed circuit board proved to be the most difficult aspect.

BIBLIOGRAPHY

- Burton M. 2012. Dallas Temperature Control Library. Address: http://milesburton.com/Main_Page?title=Dallas_Temperature_Control_Library. Accessed 29 November 2012.
- Freescale Semiconductor. 2012. Freescale Kinetis L Series Microcontrollers Now Broadly Available. Address: <http://media.freescale.com/phoenix.zhtml?c=196520&p=irol-newsArticle&ID=1737915>. Accessed 29 November 2012.
- Lajart, J. 2009. Taking an Open-Source Approach to Hardware. Wall Street Journal - Eastern Edition. 27 November 2009, Vol. 254 Issue 126, pB8.
- Malpartida F. 2012. LCD Library. Address: <https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home>. Accessed 29 November 2012.
- Massimo, B. 2008. Getting Started with Arduino. O'Reilly Media.
- Maxim Integrated. 2003. Charlieplexing – Reduced Pin-Count LED Display Multiplexing. Address: <http://www.maximintegrated.com/app-notes/index.mvp/id/1880>. Accessed 29 November 2012.
- STMicroelectronics. 2012. STMicroelectronics' STM32 Discovery Kit Establishes New Industry Standard for Low-Cost Tools for 32-bit Microcontrollers. Address: http://www.st.com/internet/com/press_release/p3065.jsp. Accessed 29 November 2012.
- Texas Instruments. 2010. TI's new \$4.30 LaunchPad is a complete development kit for harnessing the ultra-low power and 16-bit performance of MSP430™ Value Line MCUs. Address: <http://newscenter.ti.com/index.php?s=32851&item=123578>. Accessed 29 November 2012.
- Texas Instruments. 2012. Texas Instruments unveils the Stellaris® LaunchPad - a fully-functional, flexible and low-price kit for ARM® Cortex™ - M4 developers. Address: <http://newscenter.ti.com/2012-09-25-Texas-Instruments-unveils-the-Stellaris-LaunchPad-a-fully-functional-flexible-and-low-price-kit-for-ARM-Cortex-M4-developers>. Accessed 29 November 2012.